Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/CA05/001537

International filing date: 30 September 2005 (30.09.2005)

Document type: Certified copy of priority document

Document details: Country/Office: CA
Number: 2479603
Filing date: 01 October 2004 (01.10.2004)

Date of receipt at the International Bureau: 01 November 2005 (01.11.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

Office de la propriété intellectuelle du Canada

Un organisme d'Industrie Canada

Canadian Intellectual Property Office

An Agency of Industry Canada

Bureau canadien des brevets

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

Canadian Patent Office

Certification

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,479,603**, on October 1, 2004, by **SURESHCHANDRA B. PATEL**, for "Sequential and Parallel Loadflow Computation for Electrical Power System".

Agent certificateur/Certifying Officer

October 20, 2005

Date

Canada

(CIPO 68)
31-03-04

OPIC CIPO

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

SEQUENTIAL and PARELLEL LOAD-FLOW COMPUTATION FOR DECOMPOSED ELECTRICAL POWER SYSTEM

ABSTRACT

Load-Flow computations are performed as a step in real-time operation/control and in on-line/off-line studies of Electrical Power Systems. Methods in this application are the best versions of many simple variants with almost similar performance. Simple variants include any possible hybrid combination of these methods. All of the methods are characterized:

1) In the use of a method of decomposing a network referred to as Suresh's diakoptics. Suresh's diakoptics involves determining a subnetwork for each node involving directly connected nodes referred to as level-1 nodes and their directly connected nodes referred to as level-2 nodes and so on. The level of outward connectivity for local solution of a subnetwork around a given node is to be determined experimentally. This is particularly true for matrix based methods such as Newton-Raphson(NR), BGGB, FSDL and SSDL methods. Subneworks can be solved by any of the known methods including Gauss and Gauss-Seidel methods. In the case of Gauss and Gauss-Seidel methods only one level of outward connectivity around each node is found to be sufficient for the formation of subnetworks equal to the number of nodes. Sometimes it is possible that a subnetwork around a node could be a part of the subnetwork around another node making it redundant requiring local solution of less than (m+k) subnetworks. The local solution of equations of each subnetwork could be iterated for experimentally determined two or more iterations. However, maximum of three iterations were fond to be sufficient.
2) Sequential solution of subnetworks involves updating solution estimate of a subnetwork for immediate use in the solution of the next and subsequent subnetworks. This is referred to as block Gauss-Seidel approach. (steps-6 and -11 in Sequential Algorithm)
3) Parallel solution involves solution of all subnetworks using available solution estimate at the start of the iteration without intermediate updating of solution estimate. It further involves initializing a vector of dimension equal to the number of nodes with each element value zero, adding solution estimates for a node resulting from different subnetworks in a corresponding vector element, counting the number of additions and storing the average element values as initial available estimate for the next iteration.. This is referred to as block Gauss approach. **Because a node could be directly connected to two or more nodes or a part of two or more subnetworks emanating from different nodes, a parallel solution iteration involves adding and taking the average of all the solution estimates or corrections obtained for a node in the parallel solution of subnetworks emanating from different nodes** (steps-2 and -11 in parallel algorithm).
4) The parallel solution algorithm affords an opportunity to have simplified parallel computer a server procesor-arrey processors architecture, where each of the array processors communicate only with server processor and commonly shared memory locations and not among themselves.(Fig.6)

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

SEQUENTIAL and PARELLEL LOAD-FLOW COMPUTATION FOR DECOMPOSED ELECTRICAL POWER SYSTEM

TECHNICAL FIELD

The present invention relates to methods of load flow computation in power flow control and voltage control for an electrical power system. It also relates to parallel computer architecture.

BACKGROUND ART AND MOTIVATION

Utility/industrial power networks are composed of many power plants/generators interconnected through transmission/distribution lines to other loads and motors. Each of these components of the power network is protected against unhealthy or alternatively faulty, over/under voltage, and/or over loaded damaging operating conditions. Such a protection is automatic and operates without the consent of power network operator, and takes an unhealthy component out of service disconnecting it from the network. The time domain of operation of the protection is of the order of milliseconds.

The purpose of a utility/industrial power network is to meet the electricity demands of its various consumers 24-hours a day, 7-days a week while maintaining the quality of electricity supply. The quality of electricity supply means the consumer demands be met at specified voltage and frequency levels without over loaded, under/over voltage operation of any of the power network components. The operation of a power network is different at different times due to changing consumer demands and development of any faulty/contingency situation. In other words healthy operating power network is constantly subjected to small and large disturbances. These disturbances could be consumer/operator initiated, or initiated by overload and under/over voltage alleviating functions collectively referred to as security control functions and various optimization functions such as economic operation and minimization of losses, or caused by a fault/contingency incident.

For example, a power network is operating healthy and meeting quality electricity needs of its consumers. A fault occurs on a line or a transformer or a generator which faulty component gets isolated from the rest of the healthy network by virtue of the automatic operation of its protection. Such a disturbance would cause a change in the pattern of power flows in the network, which can cause over loading of one or more of the other components and/or over/under voltage at one or more nodes in the rest of the network. This in turn can isolate one or more other components out of service by virtue of the operation of associated protection, which disturbance can trigger chain reaction disintegrating the power network.

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

Therefore, the most basic and integral part of all other functions including optimizations in power network operation and control is security control. Security control means controlling power flows so that no component of the network is over loaded and controlling voltages such that there is no over voltage or under voltage at any of the nodes in the network following a disturbance small or large. Security control functions or alternatively overloads alleviation and over/under voltage alleviation functions can be realized through one or combination of more controls in the network. These involve control of power flow over tie line connecting other utility network, turbine steam/water/gas input control to control real power generated by each generator, load shedding function curtails load demands of consumers, excitation controls reactive power generated by individual generator which essentially controls generator terminal voltage, transformer taps control connected node voltage, switching in/out in capacitor/reactor banks controls reactive power at the connected node. Such overload and under/over voltage alleviation functions produce changes in controlled variables/parameters in step-60 of Fig.1. In other words controlled variables/parameters are assigned or changed to the new values in step-60. This correction in controlled variables/parameters could be even optimized in case of simulation of all possible imaginable disturbances including outage of a line and loss of generation for corrective action stored and made readily available for acting upon in case the simulated disturbance actually occurs in the power network. In fact simulation of all possible imaginable disturbances is the modern practice because corrective actions need be taken before the operation of individual protection of the power network components.

Control of an electrical power system involving power-flow control and voltage control is performed according to the process flow diagram of fig.1. The various numbered steps in fig.1 are explained in the following.

Step-10: On-line readings of open/close status of all switches and circuit breaker are obtained

Step-20: On-line readings of real and reactive power assignments or settings at PQ-nodes, real power and voltage magnitude assignments or settings at PV-nodes and transformer turns ratios are obtained. These assigned/set values are the controlled variables/parameters.

Step-30: Various power flows, voltage magnitudes and angles, reactive power generations by generators and turns ratios of transformers in the power system are determined by performing load-flow computation, which incorporates assigned/set values, in step-20 or previous process cycle step-60, of controlled variables/parameter.

Step-40: The results of Load-Flow computation of step 30 are evaluated for any over loaded transmission lines and over/under voltages at different nodes in the power system

Step-50: If the system state is good implying no over loaded lines and no over/under voltages, the process branches to step-70, otherwise to step-60

Step-60: Changes the controlled variables/parameters set in step-20 or later set in the previous process cycle step-60 and returns to step-30

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

Step-70: Actually implements the correction in controlled variables/parameters to obtain secure/correct/acceptable operation of power system

It is obvious that Load-Flow computation is performed many times in real-time operation and control environment and, therefore, **efficient and high-speed Load-Flow computation is necessary** to provide corrective control in the changing power system conditions including an outage or failure. Moreover, the **load-flow computation must be highly reliable to yield converged solution under wide range of system operating conditions and network parameters.** Failure to yield converged load-flow solution creates blind spot as to what exactly could be happening in the network leading to potentially damaging operational and control decisions/actions in capital-intensive power utilities.

The embodiment of the present invention, the most efficient and reliable loadflow computations, as described in the above steps and in Fig.1 is very general and elaborate. The control of voltage magnitude within reactive power generation capabilities of electrical machines including generators, synchronous motors, and capacitor/inductor banks, and within operating ranges of transformer taps is normally integral part of load flow computation as described in "LTC Transformers and MVAR violations in the Fast Decoupled Load Flow, IEEE PAS-101, No.9, PP. 3328-3332." If under/over voltage still exists in the results of load flow computation, other control actions are taken in step-60 in the above and in Fig.1. For example, under voltage can be alleviated by shedding some of the load connected.

However, the simplest embodiment of the efficient and reliable method of loadflow computation is where only voltage magnitudes are controlled by controlling reactive power generation of generators and synchronous motors, switching in/out in capacitor/inductor banks and transformer taps. Of course, such control is possible only within reactive power capabilities of machines and capacitor/reactor banks, and within operating ranges of transformer taps. This is the case of a network in which the real power assignments have already been fixed and where steps-50 and -60 in the above and in Fig.1 are absent. Once loadflow computations are finished, the load flow solution includes indications of reactive power generation at generator nodes and at the nodes of the capacitor/inductor banks, and indications of transformer tap settings. Based on the known reactive power capability characteristics of the individual machines, the determined reactive power values are used to adjust the excitation current to each machine, or at least each machine presently under reactive power or VAR control, to establish the desired reactive power set points. The transformer taps are set in accordance with the tap setting indications produced by the load flow computations.

This procedure can be employed either on-line or off-line. In off-line operation, the user can simulate and experiment with various sets of operating conditions and determine reactive power generation and transformer tap settings requirements. A general-purpose computer can implement the entire system. For on-line operation, the load flow computation system is provided with data identifying the current real and reactive power assignments and transformer transformation ratios, the present status of all switches and

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

circuit breakers in the network and machine characteristic curves in steps-10 and -20 in the above and in Fig. 1, and blocks 12, 20, 46, and 52 in Fig 2. Based on this information, a model of the system based on gain matrices of any of the invented or prior art load flow computation methods provide the values for the corresponding node voltages, reactive power set points for each machine and the transformation ratio and tap changer position for each transformer.

The present invention relates to control of utility/industrial power networks of the types including plurality of power plants/generators and one or more motors/loads, and connected to other external utility. In the utility/industrial systems of this type it is the usual practice to adjust the real and reactive power produced by each generator and each of the other sources including synchronous condensers and capacitor/inductor banks, in order to optimize the real and reactive power generation assignments of the system. Healthy or secure operation of the network can be shifted to optimized operation through corrective control produced by optimization functions without violation of security constraints. This is referred to as security constrained optimization of operation. Such an optimization is described in the United States Patent Number: 5,081,591 dated Jan. 13, 1992: "Optimizing Reactive Power Distribution in an Industrial Power Network", where the present invention can be embodied by replacing the block nos. 56 and 66 each by a block of constant matrices [$Y\theta$] and [YV], and replacing blocks of "Exercise Newton-Raphson Algorithm" by blocks of "Exercise Fast Super Decoupled Algorithm" in places of blocks 58 and 68. **This is just to indicate the possible embodiment of the present invention in optimization functions like in many others including state estimation function. However, inventions are claimed through a simplified embodiment without optimization function as in Fig. 2 in this application.**

DISCLOSURE OF THE INVENTION

This invention relates to steady-state power network computation referred to as Load-Flow or Power-Flow. Load-Flow computations are performed as a step in real-time operation/control and in on-line/off-line studies of Electrical Power Systems. The present invention involves two class of methods involving sequential and parallel algorithms. While the sequential algorithm yields reliable solution methods, the parallel algorithm using Suresh's diakoptics enables simplified parallel computer architecture. These sequential and parallel algorithms are the best versions of many simple variants with almost similar performance. Simple variants include any possible hybrid combination of sequential and parallel algorithmic steps.

BRIEF DESCRIPTION OF DRAWINGS

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

Fig. 1 is a flow-chart of the overall controlling method for an electrical power system involving load-flow computation as a step which can be executed using one of the sequential or parallel load-flow computation algorithms of Figs. 1 or 2.

Fig. 2 is a flow-chart of the simple special case of voltage control method in overall controlling method of Fig. 1 for an electrical power system

Fig. 3 is a flow-chart of the prior art method

Fig. 4 is a flow-chart embodiment of the invented sequential algorithm

Fig. 5 is a flow-chart embodiment of the invented parallel algorithm

Fig. 6 is a block diagram of invented parallel computer architecture

not included & to be provided on later dates

Fig. 7 is the level-1 connectivity diagrams for different nodes forming subnetworks

SYMBOLS

The prior art and inventions will now be described using the following symbols:

$\overline{Y}_{pq} = G_{pq} + jB_{pq}$	: (p-q) th element of nodal admittance matrix without shunts
$\underline{y}_p = g_p + jb_p$	: total shunt admittance at any node-p
$\overline{V}_p = e_p + jf_p = V_p \angle \theta_p$	: complex voltage of any node-p
$\Delta\theta_p$, ΔV_p	: voltage angle, magnitude corrections
Δe_p, Δf_p	: real, imaginary components of voltage corrections
$P_p + jQ_p$	: net nodal injected power calculated
$\Delta P_p + j\Delta Q_p$	: nodal power residue (mismatch)
$RP_p + jRQ_p$	: modified nodal power residue
Φ_p	: rotation angle
m	: number of PQ-nodes
k	: number of PV-nodes
n=m+k+1	: total number of nodes
q>p	: q is the node adjacent to node-p excluding the case of q=p
[ ]	: indicates enclosed variable symbol to be a vector or a matrix
LRA	: Limiting Rotation Angle
PQ-node	: load-node (Real-Power-P and Reactive-Power-Q are specified)
PV-node	: generator-node (Real-Power-P and Voltage-Magnitude-V are specified)

Prior Art

Prior art methods are described in detail in the following published documents, the description of which is avoided in this initial application. The description will be provided when this application is completed in about a year time.

1. Stagg G.W. and El-Abiad A.H., "Computer methods in Power System Analysis", McGrow-Hill, New York, 1968
2. S.B.Patel, "Fast Super Decoupled Loadflow", IEE proceedings Part-C, Vol.139, No.1, pp. 13-20, January 1992

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

3. Intenational Patent Application Number: PCT/CA/2003/001312 dated 29 August, 2003: "System of Super Super Decoupled Loadflow Computation for Electrical Power System"

Invention

Invented methods use prior art methods in Sequential and Parellel Algorithms using invented network decomposition technique referred to as Suresh's diakoptics described in the following.

Network Decomposition Technique (Suresh's Diakoptics)

A network decomposition method referred to as Suresh's diakoptics involves determining a subnetwork for each node involving directly connected nodes referred to as level-1 nodes and their directly connected nodes referred to as level-2 nodes and so on. The level of outward connectivity for local solution of a subnetwork around a given node is to be determined experimentally. This is particularly true for matrix based methods such as Newton-Raphson(NR), BGGB, FSDL and SSDL methods. Subneworks can be solved by any of the known methods including Gauss and Gauss-Seidel methods. In the case of Gauss and Gauss-Seidel methods only one level of outward connectivity around each node is found to be sufficient for the formation of subnetworks equal to the number of nodes. Sometimes it is possible that a subnetwork around any given node could be a part of the subnetwork around another node making it redundant needing local solution of less than (m+k) subnetworks. The local solution of equations of each subnetwork could be iterated for experimentally determined two or more iterations. However, maximum of three iterations were fond to be sufficient by preliminary experimentation.

It should be noted that no two decomposed network parts contain the same set of nodes, though some same nodes could appear in two or more subnetworks.

The decomposition method is further demonstrated for IEEE 14-node network in Fig.1, which shows decomposed network parts for level-1 connectivity and for level-2 connectivity.

Computation steps for Sequential Algorithm:

1. Read system data and assign an initial approximate solution. If better solution estimate is not available, set specified voltage magnitude at PV-nodes, 1.0 p.u. voltage magnitude at PQ-nodes, and all the node angles equal to that of the slack-node angle, which s is referred to as the **flat-start**
2. Form nodal admittance matrix, and Initialize iteration count ITR = 0
3. Initialize with '0' an integer vector Node(I) where I takes values from1 to n. After having determined outward level of connectivity for the formation of a subnetwork around a given node, identify the node numbers around which non redundunt subnetworks can be formed, and make corresponding elements '1' in the vector Node(I)

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

4. In case of NR, FSDL and SSDL methods form iteration matrix(ces) for each of the subnetworks, factorize them and store them
5. Store available current solution estimate for all the nodes of a network
6. **Do 11 I=1,m+k**
7. **If (Node(I) .EQ. 0) Goto 11**
8. **Do 11 J=1,N(maximum number of local iterations)**
9. **Use any of the Gauss, Gauss-Seidel, NR, BGGB, FSDL, SSDL methods to solve for complex nodal voltage or nodal voltage angle and magnitude or their corrections at the nodes of a subnetwork emanating outward from node-I.**
10. **Update complex nodal voltages or nodal votage angles and magnitudes at the nodes of subnetwork emanating from node-I**
11. **CONTINUE**
12. Increament the iteration count ITR=ITR+1
13. Determine changes in the new solution estimate from the stored one. If all the changes are not less than specified tolerance, return to step-5. Otherwise follow the next step-14
14. Calculate line flows and output the desired results

In cases of NR, BGGB, FSDL and SSDL methods convergence check is to see that real and reactive power mismatches at all nodes are less than specified tolerance. The corresponding steps will be different from the steps in the above algorithm. The sequential algorithm, which is the block Gauss-Seidel approach, has been found to increase reliability of getting converged solution using any of the methods.

Computation steps for Parallel Algorithm:

Data read and data generated in first four steps are stored in common storage to all processors including Main or Server processor where first three steps are carried out.

1. Read system data and assign an initial approximate solution. If better solution estimate is not available, set specified voltage magnitude at PV-nodes, 1.0 p.u. voltage magnitude at PQ-nodes, and all the node angles equal to that of the slack-node angle, which is referred to as the **flat-start;** or set all the nodes voltage magnitudes and angles equal to those of the slack-node, which is referred to as the **slack-start.** Use **flat-start** for Gauss and Gauss-Seidel methods, and use **slack-start** for Newton-Raphson, BGGB, FSDL and SSDL methods
2. **Initialize with '0' an integer vector Node(I) where I takes values from 1 to n. After having determined outward level of connectivity for the formation of a subnetwork around a given node, identify the node numbers around which non redundant subnetworks can be formed, and make corresponding elements '1' in the vector Node(I). Assign a processor for each node whose element value is '1' in the vector Node(I).**
3. Initialize integer vector ICOUNT(I)=0, where I takes values from 1 to n, which counts the number of times nodal variable or nodal variable corrections gets added to zero initialized nodal variable vector or nodal variables correction vectors

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

4. Initialize vectors say, complex voltage vector V1(I) in the case of Gauss and Gauss-Seidel methods, and real voltage angle vector BA(I) and real voltage magnitude vector BV(I) in case of NR, BGGB, FSDL, SSDL methods by initial solution estimate available
5. Initialize iteration count ITR = 0.

NOTE: Steps 6 to 11 are carried out in parallel using invented/available parallel computer architecture. In step-11 each processor access common to all processors memory storage.

6. Form nodal admittance matrix for each subnetwork
7. In case of Newton-Raphson, FSDL and SSDL methods, form, factorize and store iteration matrix(ces) for each subnetwork locally in a processor

NOTE: A single iteration matrix involves a node and its directly connected nodes and their directly connected nodes and their directly connected nodes and so on depending on the level of connectivity desired for a given node or optimum connectivity determined experimentally for any given network.

8. Initialize nodal variable vectors say, complex voltage vector V2(I)=0.0, real bus voltage angle correction vector DBA(I)=0.0, and real bus voltage magnitude correction vector DBM(I)=0.0 in each subnetwork in parallel
9. Iterate step 9 for a number of local iterations for each subnetwork only in case when Gauss and Gauss-Seidel methods are used
10. Use any of the Gauss, Gauss-Seidel, NR, BGGB, FSDL, SSDL methods to solve each subnetwork for complex nodal voltage or nodal voltage angle and magnitude or their corrections at the nodes of each subnetwork.
11. **Add complex nodal voltages or nodal voltage angles and magnitudes corrections at all nodes of a subnetwork to corresponding elements of nodal variable vector(s) or correction vectors, in the common storage accessible by all the processors, which were initialized zero in step-8. This is done for all subnetworks. Increament the elements of counter vector ICOUNT(I) corresponding to nodes where addition takes place. The new solution estimate vector is obtained by taking the average by dividing each element of the nodal variable vector(s) or correction vectors by the corresponding element of the vector ICOUNT (I).**
12. Update solution vectors giving new solution estimate in case of NR, BGGB, FSDL, SSDL methods.
13. Determine changes in the new solution estimate from the stored one. If all the changes are not less than specified tolerance follow the next step. Otherwise follow the step-15.
14. Change vector V1(I) in case of Gauss and Gauss-Seidel methods, or vectors BA(I) and BV(I) in case of NR, BGGB, FSDL, SSDL methods by current solution estimate available. Increament the iteration count ITR=ITR+1, and return to step-8
15. Calculate line flows and output the desired results

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

In cases of NR, BGGB, FSDL and SSDL methods convergence check is to see that real and reactive power mismatches at all nodes are less than specified tolerance. The corresponding steps will be different from the steps in the above algorithm.. **The parallel algorithm described in the above, which is the block Gauss approach,** has been found to increase efficiency and reliability of getting converged loadflow solution. There could be variations in steps in the above algorithm. However, **the main inventive steps being claimed are steps 2 and 11. Step-2 helps decompose a given network in subnetworks or alternatively blocks of network, and step-11 updates a complete network nodal variables without intermediate updating creating a block Gauss approach.**

The system stores a representation of the reactive capability characteristic of each machine and these characteristics act as constraints on the reactive power, which can be calculated for each machine.

While the description above refers to particular embodiments of the present invention, it will be understood that many modifications may be made without departing from the spirit thereof. The accompanying claims are intended to cover such modifications as would fall within the true scope and spirit of the present invention.

The presently disclosed embodiments are therefore to be considered in all respect as illustrative and not restrictive, the scope of the invention being indicated by the appended claims in addition to the foregoing description, and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

CLAIMS

The present invention is applicable to systems to process load flow computation by means of block Gauss-Seidel approach using sequential algorithm and block Gauss approach using parallel algorithm on invented (Fig.6)/available computer of parallel computer architecture. The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A method of controlling voltages and power flows in a power network, comprising the steps of:

   obtaining on-line data of open/close status of switches and circuit breakers in the power network,
   obtaining on-line readings of real and reactive power assignments or settings at PQ-nodes, real power and voltage magnitude assignments or settings at PV-nodes and transformer turns ratios, which are the controlled variables/parameters,
   performing load-flow computation using Suresh's diakoptical method for breaking a network into subnetworks and solving them by a sequential parallel algorithm,
   evaluating the computed load flow for any of the over loaded power network components and for under/over voltage at any of the nodes,
   correcting one or more controlled parameters and repeating the computing and evaluating steps until evaluating step finds a good power system without any over loaded components and without any under/over voltages in the power network, and
   effecting a change in the power flowing through network components and voltages and phases at the nodes of the power network by actually implementing the finally obtained values of controlled parameters after evaluating step finds a good power system.

2. A load-flow computation as defined in claim1 is characterized in the use of a method of decomposing a network referred to as Suresh's diakoptics. Suresh's diakoptics involves determining a subnetwork for each node involving directly connected nodes referred to as level-1 nodes and their directly connected nodes referred to as level-2 nodes and so on. The level of outward connectivity for local solution of a subnetwork around a given node is to be determined experimentally. This is particularly true for matrix based methods such as Newton-Raphson(NR), BGGB, FSDL and SSDL methods. Subneworks can be solved by any of the known methods including Gauss and Gauss-Seidel methods. In the case of Gauss and Gauss-Seidel methods only one level of outward connectivity around each node is found to be sufficient for the formation of subnetworks equal to the number of nodes. Sometimes it is possible that a subnetwork around a node could be a part of the subnetwork around another node making it

Inventor: Sureshchandra B. Patel
New Address: 159 Campbell Avenue, Toronto, Ontario M6P 3V3
E-mail: trruth@hotmail.com

redundant requiring local solution of less than (m+k) subnetworks. The local solution of equations of each subnetwork could be iterated for experimentally determined two or more iterations. However, maximum of three iterations were fond to be sufficient.

3. A load-flow computation as defined in claim1 is characterized in the use of sequential solution of subnetworks involving updating solution estimate of a subnetwork for immediate use in the solution of the next and subsequent subnetworks. This is referred to as block Gauss-Seidel approach. (steps 6 to 11 in Sequential Algorithms)

4. A load-flow computation as defined in claim1 is characterized in the use of Parallel solution involving solution of all subnetworks using available solution estimate at the start of the iteration without intermediate updating of solution estimate. It further involves initializing a vector of dimension equal to the number of nodes with each element value zero, adding solution estimates for a node resulting from different subnetworks in a corresponding vector element, counting the number of additions and storing the average element values as initial available estimate for the next iteration.. This is referred to as block Gauss approach. **Because a node could be directly connected to two or more nodes or a part of two or more subnetworks emanating from different nodes, a parallel solution iteration involves adding and taking the average of all the solution estimates or corrections obtained for a node in the parallel solution of subnetworks emanating from different nodes** (steps-2 and -11 in parallel algorithm).

5. A load-flow computation as defined in claim1 and claim 4 is characterized in the use of the simplified parallel computer a server processor-array processors architecture for the solution parallel solution algorithm, where each of the array processors communicate only with server processor and commonly shared memory locations and not among themselves. (Fig.6)

Inventor: Sureshchandra Patel
Patent Application Number: New **Sheet 1**

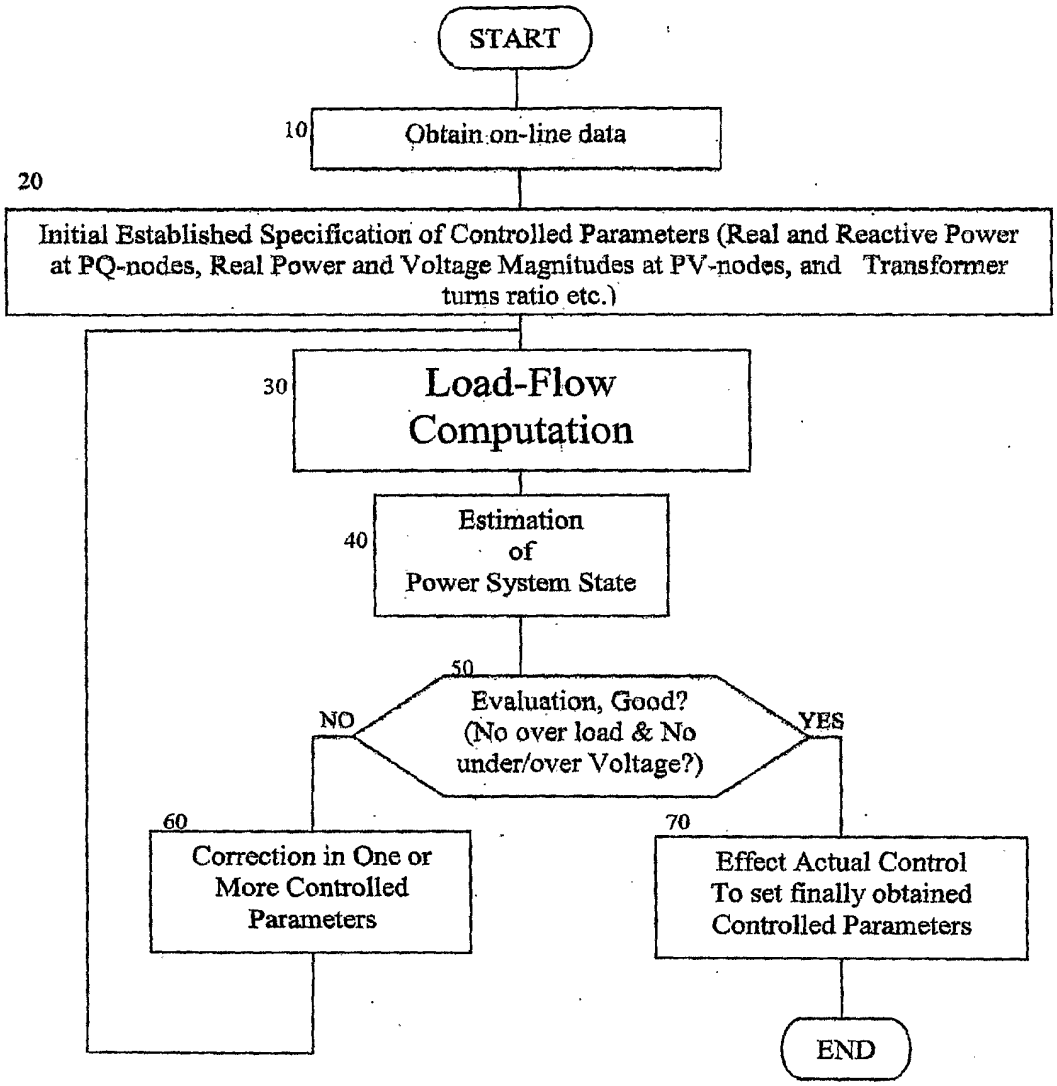

Fig. 1: Load-Flow Computation in Security Control of Electrical Power System (PRIOR ART)

Inventor: Sureshchandra Patel
Patent Application Number: New

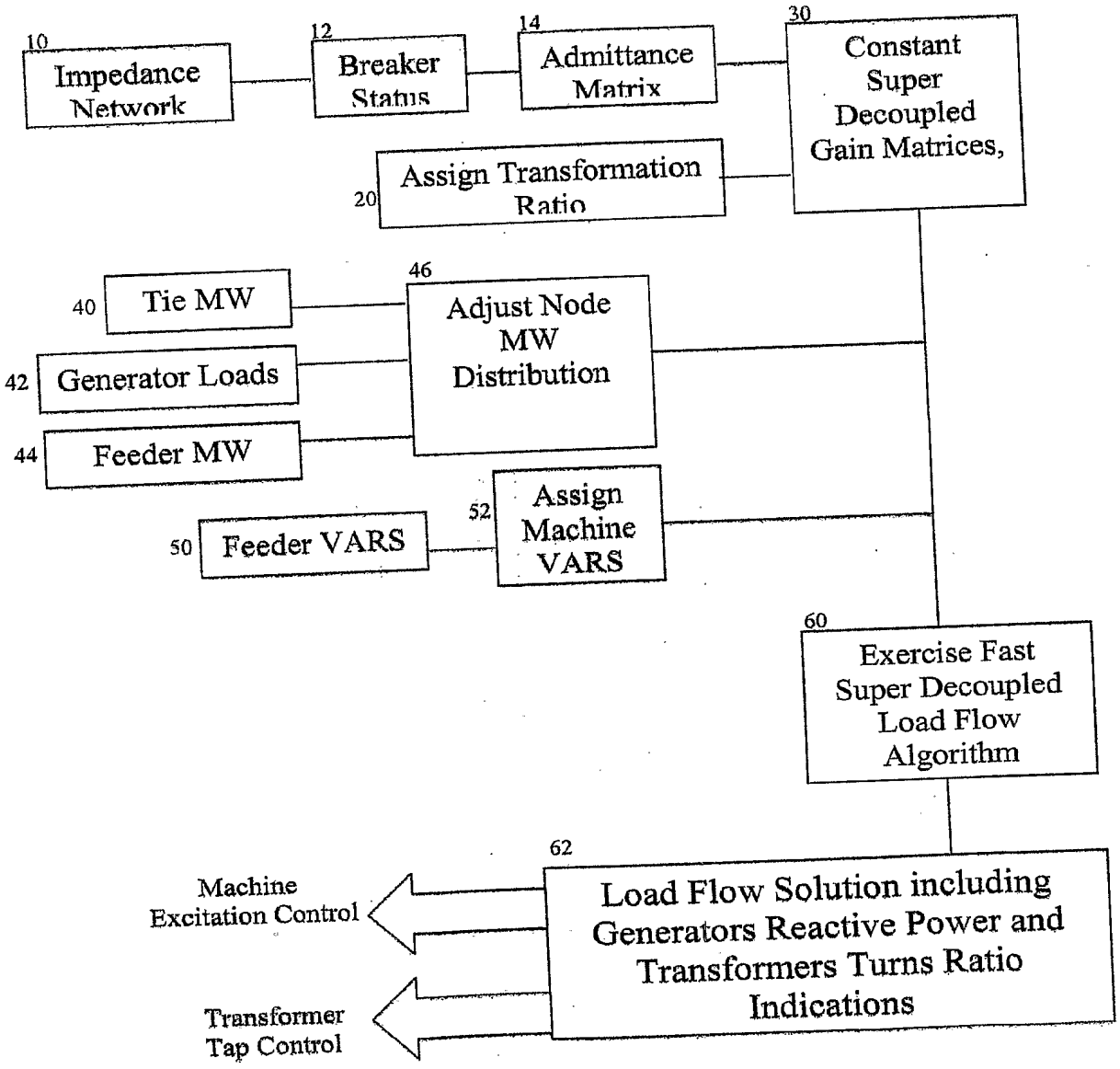

Fig. 2: Load-Flow Computation in Voltage Control of Electrical Power System (PRIOR ART)

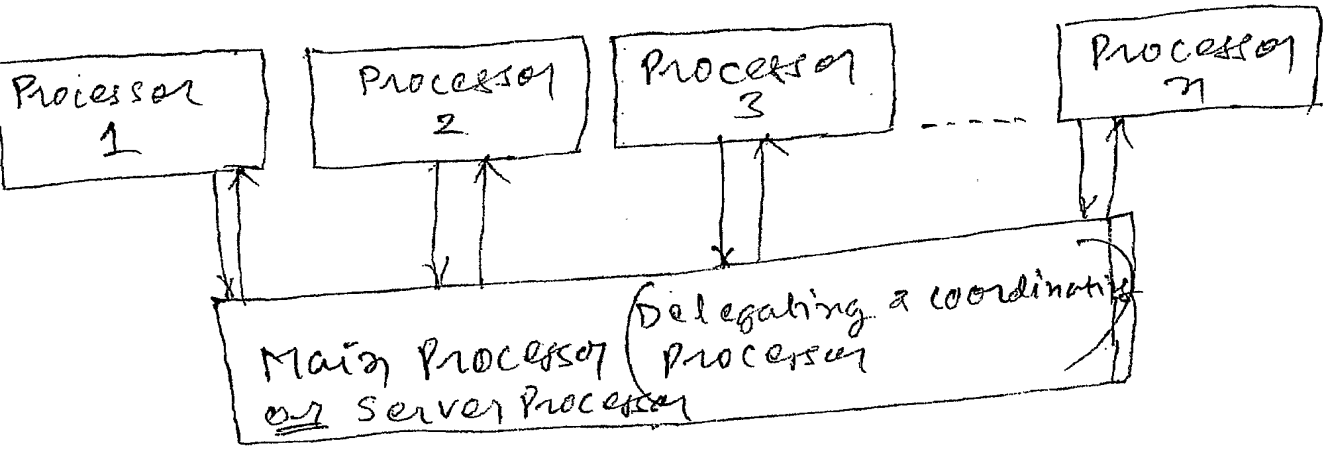

Fig. 6 Parallel Processor Architecture which constitute the CPU of a personal computer

Each processor has its own processing and temporary storage memory in addition to common memory shared by all processors and server processor.

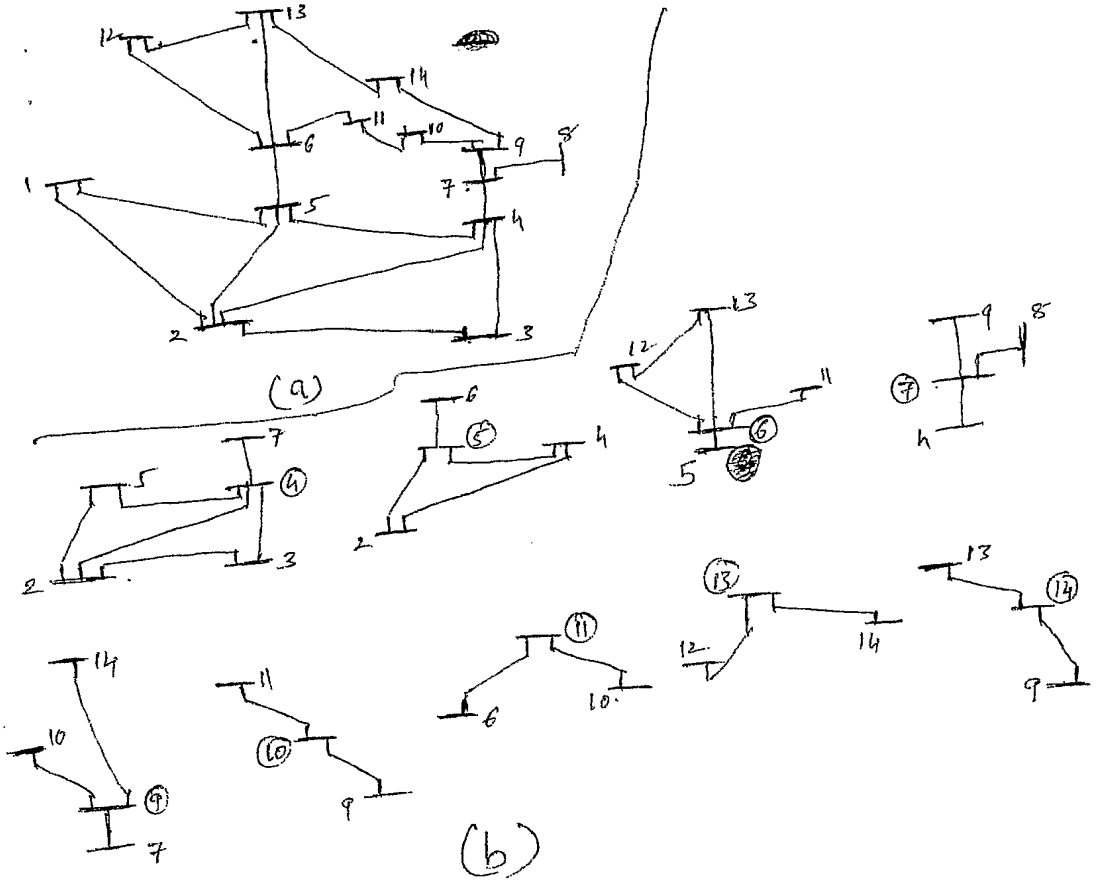

Fig: 7

(a) connection Diagram of IEEE-14 node system

(b) connection diagrams (Level-1) for different nodes enclosed in circle.

Note: Connection diagrams for nodes ② and ③ are part of level-1 connection diagrams of node ①, and which are redundant similarly, level-1 connection diagrams for nodes ⑥ and ⑫ are redundant.